39. [Amended] The cache coherency system of claim 37, said memory controller further being selectively operable for sending cache update messages to update corresponding cache lines at all remote nodes having copies of a changed cache line and for receiving cache lines of data from remote nodes for updating the cache at said first node.

--

REMARKS

The above amendment and these remarks are responsive to the Office Action of 19 Jan 2001 by Examiner Pierre-Michel Bataille.

Claims 1 and 31-39 are in the case, none having been allowed.

Information Disclosure Statement

Applicant expresses appreciation for the Examiner's consideration of the U.S. patent references noted on the

information disclosure statement submitted on 19 Sep 1999, paper No. 3, and encloses herewith a copy of the non-patent reference, Michel DuBois, Effects of Cache Coherency in Multiprocessor, Transacitons on Computers, Vol. c-31, No. 11, Nov 1982, pp. 1083-1099.

Drawings

Applicant will submit formal drawings when the application is allowed.

Response to Amendment

The amendment filed September 10, 1999 has not been fully entered, with the explanation: "..the amendment requests correction on page 1, line 2. The Text to be replaced is not included on the noted page or line."

The amendment filed 10 Sep 1999 was in error, and has been corrected.

Specification

The specification has been objected to, with the requirement that the status of copending and divisional applications be updated. Applicant has amended the specification as required.

35 U.S.C. 112

Claims 37, 38 and 39 have been rejected under 35 U.S.C. 112.

Applicant has amended claim 37 to clarify as requested, and has amended claims 38 and 39 to depend from claim 37, thus to provide antecedent basis for the recitation of "said memory controller".

35 U.S.C. 102

Claims 1, and 31-36 have been rejected as being anticipated by Gupta et al.(5,535,116; hereafter, Gupta).

Applicant agrees that Gupta teaches a directory-based cache coherency system for a shared memory parallel processing system. However, the present invention teaches a different type of cache coherency system than Gupta, as follows:

1) **Attraction Memory vs. Physical Memory**

Gupta teaches a system using Attraction Memory, where the shared memory is a cache. Gupta does not teach a cache coherency system for physical memory. The present invention applies to physical shared memory (shared memory is a memory that is neither a cache nor an attraction memory).

Gupta (Col. 3, lines 54-58) states: "The main memory portion is an attraction memory cache. The attraction memory cache is preferably divided into a plurality of banks. In addition, the attraction memory is preferably a set-associative attraction memory. The attraction memory cache is not backed-up by physical memory."

Gupta's attraction memory approach means that there is no physical portion of memory associated with each node, as in the present invention. Instead, only the addresses of

physical shared memory are allocated to each node, although the corresponding contents of the addressed memory location is probably not located at that node. Gupta explains this in (Col.3, line 27-32):

> "In particular, the directory memory has a state memory for maintaining the current state of each data item assigned to the respective processing node. In addition, the directory has a pointer to a master node for each data item, the master node being the exactly one processing node having a master copy of the data item."

The present invention teaches a section of physical memory assigned to each node and the corresponding directory associated with tracking that section of memory being located at the same node. (Page 17, lines 10-12 and Page 19, line 28 to Page 20.) The present invention does not have a master copy located at a different node from the directory tracking that data item.

2) Home Node and Master Node vs. Physical Node

Gupta explains his approach in (Col.3, line 56 to col. 4, line 2):

> "A first access request is transmitted from the local processing node to a home processing node. The home processing node is statically assigned to the data item. In the home processing node, a master processing node is determined by a home directory. The master processing node has a current copy of the data item. A second access request is transmitted from the home processing node to the master processing node."

Since each cache data line does not usually reside at the home node which is tracking it, Gupta typically requires four network messages as follows to access any cache line for a requesting node. 1) The requesting node sends the read request over the network to the home node first. The home node access its directory to find the "master" node; i.e., the node which has the master copy of the requested data. 2) The home node forwards the request to the master node. 3) The master node returns a copy of the requested data over the network to the requesting node. 4) The

requesting node then sends an acknowledgment message to the home node to verify that it has received the requested data (Col. 11, lines 17-25). The acknowledgment message is necessary for Gupta's method to work, as explained in Col. 11, lines 41-52). The home node records in its directory that the requesting node has a copy of the data line.

The present invention differs in that it is more efficient, having statically assigned shared memory as physical memory at each node and requiring only two network messages to access any cache line. 1) A read request goes to the node implementing the shared memory location, and 2) the data is accessed and returned while the directory is updated in parallel (applicant's specification, page 9, lines 20-26). Also, the present invention doesn't need or require the acknowledgment message to function properly, thus saving time and network congestion.

Claims 1 and 31 have been amended to clarify the distinctions with respect to Gupta, and claims 35-39 depend from claim 1. Applicant requests that these claims be allowed.

35 U.S.C. 103

Claims 37-39 have been rejected under 35 U.S.C. 103(a) over Gupta in view of Hagersten et al. (U.S. Patent 5,887,138).

Claims 37-39 depend from claim 1 which has been amended to clarify the distinctions with respect to Gupta, as previously discussed. Hagersten is similarly distinguished, for not teaching the use of portion of shared memory which is not a cache, as set forth in the independent claims of the present application.

SUMMARY AND CONCLUSION

Attached hereto is a marked-up version of the changes made to the specification and claims by the current amendment. The attachment is captioned **"Version with markings to show changes made."**

Applicants urge that the above amendments be entered and the case passed to issue with claims 1 and 31-39.

If, in the opinion of the Examiner, a telephone conversation with applicant(s) attorney could possibly facilitate prosecution of the case, he may be reached at the number noted below.

Sincerely,

Howard T. Olnowich

By

Shelley M Beckstrand
Reg. No. 24,886

Date: 18 April 2001

Shelley M Beckstrand, P.C.
Attorney at Law
314 Main Street
Owego, NY 13827

Phone: (607) 687-9913
Fax: (607) 687-7848

VERSION WITH MARKINGS TO SHOW CHANGES MADE

In the Specification

Paragraph beginning at line 2 of page 12 has been amended as follows:

-- FIGs. 1A and 1B, arranged as shown in FIG. 1 [is] are a diagram of a typical digital network showing the interconnection of a network node to the network and the components of the network node according to the preferred embodiment of this invention.--

Paragraph inserted after line 3 of page 1 has been amended as follows:

-- This application is a divisional of U.S. Patent Application S/N 08/890,341 filed 10 Jul 1997 by Howard T. Olnowich for Memory Controller for Controlling Memory Accesses Across Networks in Distributed Shared Memory Processing Systems (as amended), now U. S. Patent 6,044,438 issued 28 March 2000, other divisions of which have issued on 19 Sep 2000 as U. S. Patents 6,122,659 and 6,122,674.--

Paragraph beginning at line 4 of page 1 has been amended as follows:

-- U.S. patent application Serial No. 08/891,404, filed 10 Jul 1997, entitled "Cache Coherent Network Adapter for Scalable Shared Memory Processing Systems", [assignee docket EN997036,] (now U.S. patent 6,092,155 issued 18 Jul 2000 and pending divisional application S/N 09/516,393 filed 1 Mar 2000) filed concurrently herewith is assigned to the same assignee hereof and contains subject matter related, in certain respects, to the subject matter of the present application; it is incorporated herein by reference.--

In the Claims

Claims 1, 31-34 and 37-39 have been amended as follows. Claims 35-36 are also included for convenience.

--

1. [Twice Amended] A cache coherency system for a shared memory parallel processing system including a plurality of processing nodes, comprising:

a multi-stage communication network for interconnecting said processing nodes;

each said processing node including a unique section of shared memory which is not a cache;

each said processing node including one or more caches for storing a plurality of cache lines; [and]

a cache coherency directory which is distributed to each of said nodes for tracking which of said nodes have copies of each cache line; and

an adapter for storing changed data immediately to said unique section of shared memory regardless of which of said nodes is changing the data and which of said nodes includes the section of shared memory to be changed, such that said shared memory always contains the most recent data.

31. [Amended] A method for operating a shared memory parallel processing system as a cache coherency system including a plurality of processing nodes, each said

processing node including a unique section of shared memory which is not a cache, comprising the steps of:

interconnecting said processing nodes through a multi-stage communication network;

storing at each said processing node a plurality of cache lines in one or more caches;

distributing to each of said processing nodes a cache coherency directory; [and]

tracking in said cache coherency directory which of said processing nodes have copies of each cache line; and

changing said shared memory, wherein changed data is stored immediately to said unique section of shared memory regardless of which of said nodes is changing the data and which of said nodes includes the section of shared memory to be changed, wherein said shared memory always contains the most recent data.

32. [Amended] A program storage device readable by a machine, tangibly embodying a program of instructions executable by a machine to perform method steps for operating a shared memory parallel processing system including a plurality of processing nodes, each said processing node including a unique section of shared memory which is not a cache, said method steps comprising:

interconnecting said processing nodes through a multi-stage communication network;

storing at each said processing node a plurality of cache lines in one or more caches; [and]

tracking in a cache coherency directory which is distributed to each of said processing nodes which of said processing nodes have copies of each cache line; and

changing said unique section of shared memory, wherein changed data is stored immediately to shared memory regardless of which of said nodes is changing the data and which of said nodes includes the section of shared memory to be changed, wherein said shared memory always

contains the most recent data.

33. [Amended] An article of manufacture comprising:

a computer useable medium having computer readable program code means embodied therein for operating a shared memory parallel processing system including a plurality of processing nodes, each said processing node including a unique section of shared memory which is not a cache, the computer readable program means in said article of manufacture comprising:

computer readable program code means for causing a computer to effect interconnecting said processing nodes through a multi-stage communication network;

computer readable program code means for causing a computer to effect storing at each said processing node a plurality of cache lines in one or more caches; [and]

computer readable program code means for causing a computer to effect tracking in a cache coherency directory which is distributed to each of said

processing nodes which of said processing nodes have copies of each cache line; and

computer readable program code means for storing changed data immediately to said unique section of shared memory regardless of which of said nodes is changing the data and which of said nodes includes the section of shared memory to be changed such that said shared memory always contains the most recent data.

34. [Amended] A computer program product or computer program element for operating a shared memory parallel processing system including a plurality of processing nodes, each said node including a unique section of shared memory which is not a cache, according to the steps of:

interconnecting said processing nodes through a multi-stage communication network;

storing at each said processing node a plurality of cache lines in one or more caches;

distributing to each of said processing nodes a cache coherency directory; [and]

tracking in said cache coherency directory which of said processing nodes have copies of each cache line; and

storing changed data immediately to said unique section of shared memory regardless of which of said nodes is changing the data and which of said nodes includes the section of shared memory to be changed such that said shared memory always contains the most recent data.

35. The cache coherency system of claim 1, further comprising:

a shared memory including a first memory portion for storing unchangeable data and a second memory portion for storing changeable data; and

said cache coherency directory listing which nodes of said plurality of processing nodes have accessed copies of said cache lines in said second memory portion.

36. The cache coherency system of claim 35, each of said plurality of processing nodes being operable for reading,

storing, and invalidating said shared memory at any other of said processing nodes.

37. [Amended] The cache coherency system of claim 36, further comprising at a first node of said plurality of processing nodes a memory controller selectively operable first responsive to a request for access to a memory word [for] by first accessing the cache at said first node and, if said requested memory word is not available in said cache, selectively operable second [responsive to being unable to access said memory word in said cache at said first node] for accessing said memory word selectively from [a cache line in said memory or from a remote] said shared memory regardless of which of said nodes includes the section of shared memory being accessed, and storing said cache line including said memory word to said cache at said first node.

38. [Amended] The cache coherency system of claim [36] 37, said memory controller further being selectively operable for deleting a cache line from said cache at said first node when said cache is full to provide space for a new cache line to be stored to said cache, and for sending the address

of the deleted cache line to an invalidation directory to indicate said node no longer has a copy of said cache line.

39. [Amended] The cache coherency system of claim [36] 37, said memory controller further being selectively operable for sending cache update messages to update corresponding cache lines at all remote nodes having copies of a changed cache line and for receiving cache lines of data from remote nodes for updating the cache at said first node.

--